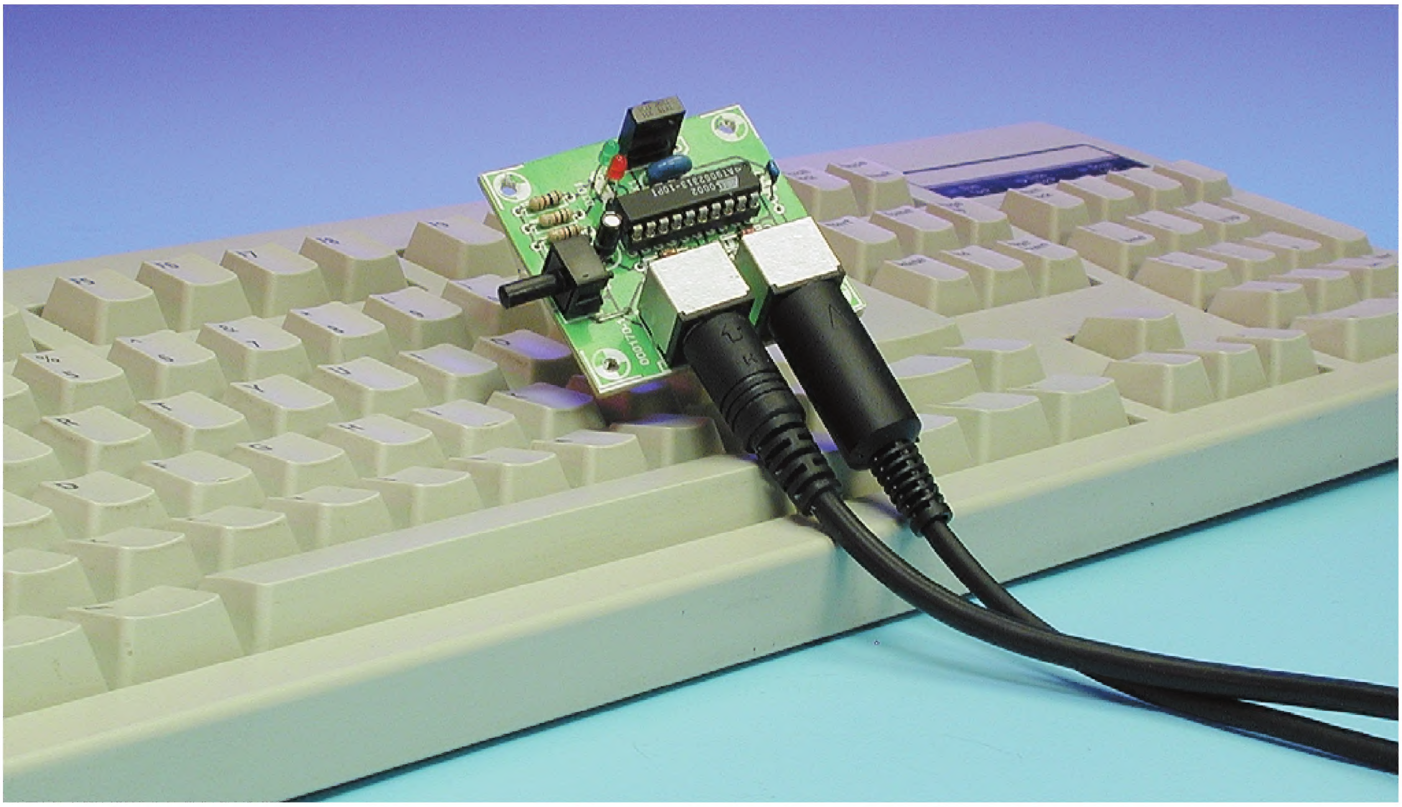
Infrared Remote Control for PCs

using the RC5 code

By J. Ferber

McFerber@gmx.net

Using this infrared receiver with your PC will let you control your favourite programs using a readily-available RC5 transmitter: useful if your PC has a suitable video card and you use it as a television, or if you want to use Winamp, for example, to play MP3 files through your hi-fi.



If you want to install a PC in your living room as a television or as part of your hi-fi setup, a remote control is practically indispensable. Who wants the trouble of going to the PC's keyboard to adjust the volume or change channels? Unfortunately, PCs do not offer built-in infrared receivers compatible with ordinary transmitters. And so we must make the necessary hard- and software ourselves.

There are several steps involved in getting a PC to react to the pulses of infrared sent out by the transmitter. Hardware is required to receive the pulses and process them into a form suitable for input to a microcontroller, where, according to how the user has programmed it, they can be translated into a different format. In principle any of the PC's interface ports can be used, for example the serial RS-232 interface or the parallel Centronics interface. Our receiver, however, is designed to work using the keyboard port. This has many advantages: it keeps the hardware to a minimum (for example, no level shifters are required); power can be drawn from the PC's keyboard connector, removing the need for an external power supply; and the software can be arranged to pass signals through from PC to keyboard and vice versa, so that the receiver can stay permanently in circuit, removing the need to keep changing plugs and sockets.

No software needs to be installed on the computer, making the circuit completely independent of operating system, working equally well under Windows, Linux or DOS. And because Apple computers use the same AT keyboard protocol as PCs, the circuit also works with MacOS.

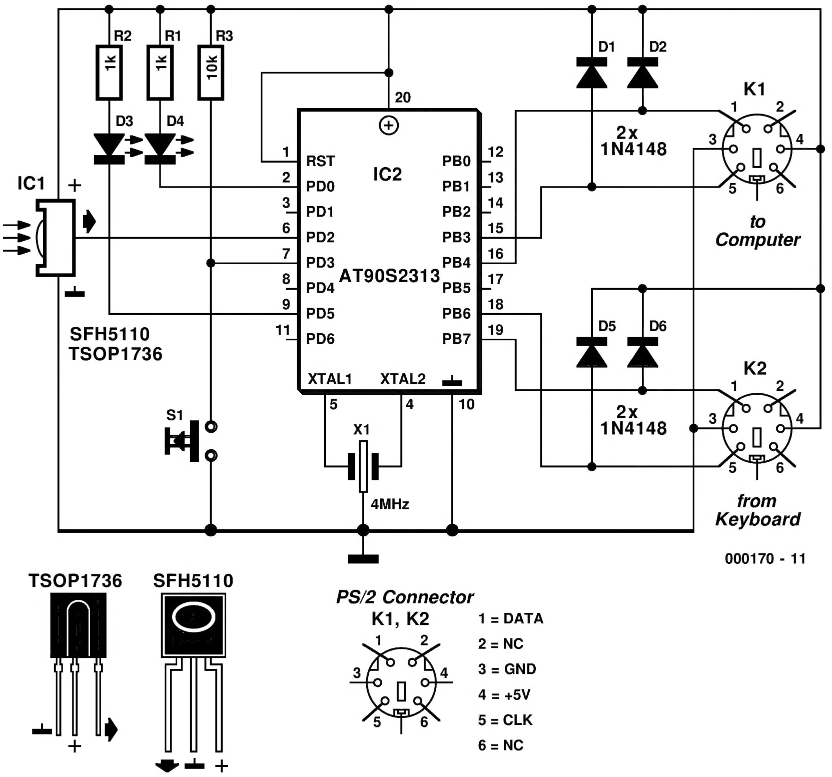


Figure 1. Circuit diagram of the remote control receiver.

Hardware and Software

As can be seen from **Figure 1**, the hardware is rather simple. There are only two ICs, a couple of passive components, and, for the 'user interface', a push-button and two LEDs. The keyboard is plugged into PS2 socket K2, and the PC is connected to PS2 socket K1 using a suitable male-to-male cable.

IC1 is the well-known combined infrared receiver/demodulator type TSOP1736 from Vishay, see url [1] or alternatively the SFH5110, the successor to the SFH506 from Siemens (Electrovalue). When selecting the component the modulation frequency of 36 kHz must be taken into account: this frequency is used in most infrared remote control systems. At the output of the receiver (pin 2) the transmitted signal is available demodulated and amplified, and with noise removed.

The microcontroller, an AT90S2313 from Atmel, receives this signal on port PD2. This device is part of Atmel's AVR family of 8-bit RISC microcontrollers and has already been seen in several Elektor Electronics projects. The instruction set, a data sheet and application notes can be found at [2].

For us, the most important characteristic of the AT90S2313 is that the microcontroller offers 128 bytes of EEPROM which can be used to record infrared codes and the corresponding keyboard commands during the programming phase, to be read out again during normal operation.

The complete program, which is around 1 kilobyte long, was written using the Atmel assembler which is freely available from their website at [3]. You can obtain the software in three ways: simplest (although most expensive) is to buy a ready-programmed microcontroller from the Publishers' Readers Services Readers' Services as order code **000170-41**. Alternatively, you can program the microcontroller yourself: the software (source and object code) is available on diskette (order code **000170-11**) or freely downloadable from the Elektor Electronics website [4]. There you will also find a PDF file containing the circuit board layout.

The present version of the software is designed only for use with RC5-compatible transmitters; however, the author is working on implementing other remote control protocols. A difficulty here is the number of control bits used (and hence the total number of infrared commands available). Since the address in the EEPROM keystroke table is derived from the infrared command code, the number of command codes may not exceed the number of bytes in the EEPROM (128 in this case). This is however the case for several infrared protocols. One solution would be to add an external I2C EEPROM device.

There is little to be said about building the tiny circuit board shown in **Figure 2**. As long as all the components are fitted correctly (don't forget that the infrared receiver's lens must point outwards!) the circuit should work straight away. It is important to connect the two PS2 connectors correctly or the software will not work (although the green LED will flash).

Beware: fitting the microcontroller or infrared receiver incorrectly may damage your motherboard: before fitting the ICs check the polarity of the supply voltage.

Programming

User programming is very straightforward. In the initial state, when the PC is switched on, the green LED flashes. Press button S1, and

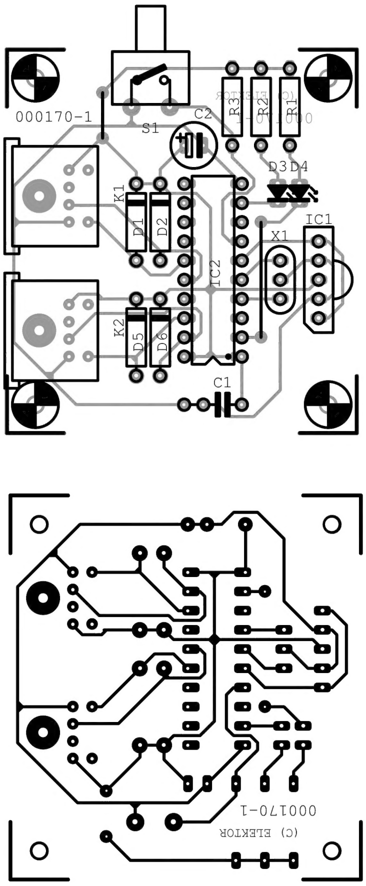
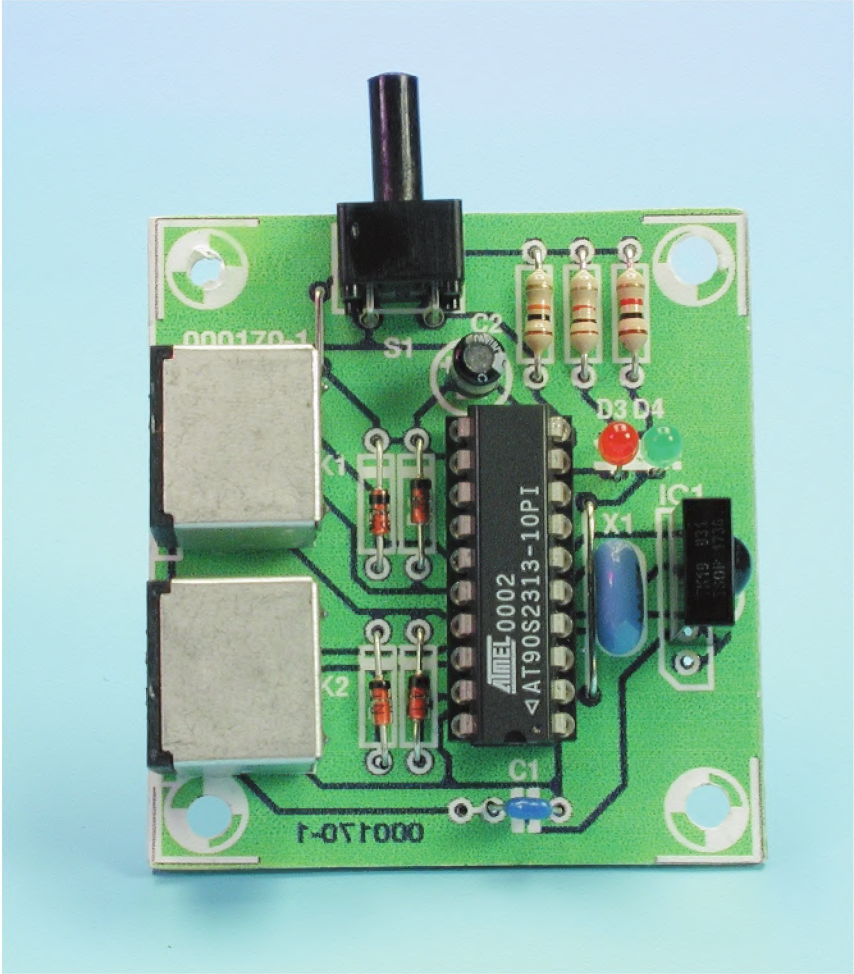


Figure 2. PCB artwork.



the microcontroller software goes into programming mode, indicated to the user by the green LED being lit steadily. Now press a button on the infrared transmitter, for example PLAY. Reception of the command is confirmed by the red LED. Now press the corresponding key on the keyboard (in Winamp the X key is PLAY) and the red LED changes state again. The command is now programmed, and can be overwritten using the same procedure.

Key combinations such as control-T are entered as usual by holding down the control key while tapping T. Further commands can be entered in the same way until the EEPROM is full, with 64 commands.

Because of the rather limited amount of EEPROM space, only single and double key codes are allowed (control-S is allowed, control-Alt-S is not). This covers most of the normal shortcut combinations. One suggestion would be to call up unsupported shortcuts or longer strings of keys as a macro using a supported shortcut.

When programming is complete, programming mode is exited by pressing the push-button again: the green LED flashes. Now if you press a button on the transmitter, the microcontroller acknowledges with a brief flash on the red LED and sends the appropriate keyboard code to the PC.

(000170-1)

COMPONENTS LIST

Resistors:

R1,R2 = 1kΩ
R3 = 10kΩ

Capacitors:

C1 = 100nF
C2 = 10µF 16V radial

Semiconductors:

D1,D2,D5,D6 = 1N4148
D3 = LED, red, high efficiency
D4 = LED, green, high efficiency
IC1 = TSOP1736 (Conrad Electronics #171069) or SFH5110 (C-I Electronics #01.00.5110.36)
IC2 = AT90S2313, programmed, order code 000170-41

Miscellaneous:

S1 = pushbutton, 1 make contact
K1,K2 = 6-way mini DIN-socket, PCB mount
X1 = Resonator, 4MHz, 3 connections
PCB, order code 000170-1
Disk, project software, order code 000170-11

Website urls

[1] www.vishay.com/products/optoelectronics/tsop17x.html

[2] www.atmel.com/atmel/products/prod200.htm

[3] www.atmel.com/atmel/products/prod203.htm

[4] www.elektor-electronics.co.uk